

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-037246

(43)Date of publication of application : 07.02.1997

(51)Int.Cl. H04N 7/24  
 B41J 2/52  
 H03M 7/40  
 H04N 1/41  
 H04N 1/415

(21)Application number : 07-178695

(71)Applicant : CANON INC

(22)Date of filing : 14.07.1995

(72)Inventor : HIRABAYASHI KOJI  
 KAJIWARA HIROSHI  
 YOSHIDA TADASHI

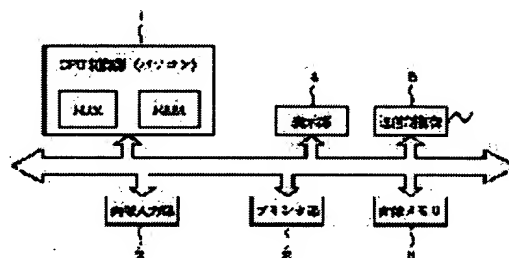
## (54) IMAGE PROCESSING UNIT AND ITS METHOD

## (57)Abstract:

**PROBLEM TO BE SOLVED:** To attain coding employing the optimum coding method depending on the property of an image by allowing a division means to divide an image in which coding object block with different size is in existence in the image.

**SOLUTION:** The processing unit is provided with a CPU control section 1 coding and decoding image data through the software processing, an image input section 2, a printer section 3, a display section 4, a communication control section 5 that sends coded data by the CPU control section 1 or receives coded data sent from an external device and an image memory 6.

Then image data stored in the image memory 6 are divided into the unit of  $128 \times 128$  picture elements as a basic block and the data are sequentially extracted for each basic block. Thus, the division method to a block in the case of coding finally the image data and the optimum coding method with respect to each block to be divided are tentatively decided. In this case, the division of each block allows the presence of coding object blocks of different size in an image.



## LEGAL STATUS

[Date of request for examination]

04.12.2000

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision  
of rejection]

[Date of requesting appeal against examiner's  
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平9-37246

(43)公開日 平成9年(1997)2月7日

(51)Int.Cl. <sup>8</sup>	識別記号	庁内整理番号	F I	技術表示箇所
H 0 4 N 7/24			H 0 4 N 7/13	Z
B 4 1 J 2/52		9382-5K	H 0 3 M 7/40	
H 0 3 M 7/40			H 0 4 N 1/41	B
H 0 4 N 1/41			1/415	
1/415			B 4 1 J 3/00	A
審査請求 未請求 請求項の数17 O L (全 10 頁)				

(21)出願番号 特願平7-178695

(22)出願日 平成7年(1995)7月14日

(71)出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72)発明者 平林 康二

東京都大田区下丸子3丁目30番2号キヤノン株式会社内

(72)発明者 梶原 浩

東京都大田区下丸子3丁目30番2号キヤノン株式会社内

(72)発明者 吉田 正

東京都大田区下丸子3丁目30番2号キヤノン株式会社内

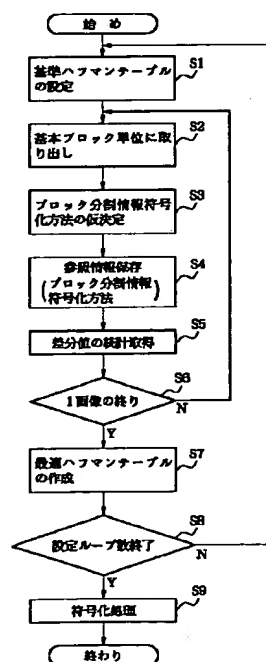
(74)代理人 弁理士 丸島 儀一

(54)【発明の名称】 画像処理装置及び方法

(57)【要約】

【課題】 画像の性質に応じて、最適な画像処理単位で最適な符号化方法を用いた符号化を行うことを目的とする。

【解決手段】 画像を複数の符号化対象ブロックに分割する分割手段と、前記分割手段により分割された各々の符号化対象ブロックに対して、複数個用意された符号化モードから前記符号化対象ブロックに用いる符号化モードを1つを選択する選択手段と、前記選択手段により選択された符号化モードに応じて前記各々の符号化対象ブロックを符号化する符号化手段とを有し、前記分割手段による分割は、前記画像内に異なるサイズの符号化対象ブロックの存在を可能とすることを特徴とする画像処理装置。



## 【特許請求の範囲】

【請求項1】 画像を複数の符号化対象ブロックに分割する分割手段と、

前記分割手段により分割された各々の符号化対象ブロックに対して、複数個用意された符号化モードから前記符号化対象ブロックに用いる符号化モードを1つを選択する選択手段と、

前記選択手段により選択された符号化モードに応じて前記各々の符号化対象ブロックを符号化する符号化手段とを有し、

前記分割手段による分割は、前記画像内に異なるサイズの符号化対象ブロックの存在を可能とすることを特徴とする画像処理装置。

【請求項2】 更に、前記画像に対し前記分割手段による分割と、前記選択手段による選択とを反復する反復手段を有することを特徴とする請求項1に記載の画像処理装置。

【請求項3】 画像を複数の符号化対象ブロックに分割する分割ステップと、

前記分割ステップで分割された各々の符号化対象ブロックに対して、複数個用意された符号化モードから前記符号化対象ブロックに用いる符号化モードを1つを選択する選択ステップと、

前記選択ステップで選択された符号化モードに応じて前記各々の符号化対象ブロックを符号化する符号化ステップとを有し、

前記分割ステップの分割は、前記画像内に異なるサイズの符号化対象ブロックの存在を可能とすることを特徴とする画像処理方法。

【請求項4】 画像を可変サイズの複数の符号化対象ブロックに分割する分割手段と、

前記分割手段により分割された複数の符号化対象ブロックに用いるハフマンテーブルを前記サイズ毎に夫々作成する作成手段と、

前記作成手段により作成された前記サイズ毎のハフマンテーブルを用いて、該サイズに対応する符号化対象ブロックを符号化する符号化手段とを有することを特徴とする画像処理装置。

【請求項5】 更に、前記画像に対し、前記分割手段による分割と、前記作成手段による作成とを反復する反復手段を有することを特徴とする請求項4に記載の画像処理装置。

【請求項6】 画像を可変サイズの複数の符号化対象ブロックに分割する分割ステップと、

前記分割ステップで分割された複数の符号化対象ブロックに用いるハフマンテーブルを前記サイズ毎に夫々作成する作成ステップと、

前記作成ステップで作成された前記サイズ毎のハフマンテーブルを用いて、該サイズに対応する符号化対象ブロックを符号化する符号化ステップとを有することを特徴

とする画像処理方法。

【請求項7】 画像を多値画像として入力する入力手段と、

前記入力手段により入力された前記多値画像をブロックに分割する分割手段と、

前記分割手段により分割されたブロック内の画素値のレベルの個数が所定の数かどうかを判別する判別手段と、

前記判別手段により、前記レベルの個数が所定の数であると判別された場合は、前記判別手段により判別されたブロックに対してランレングス符号化を行う符号化手段とを有することを特徴とする画像処理装置。

【請求項8】 前記ブロックは2次元ブロックであることを特徴とする請求項7に記載の画像処理装置。

【請求項9】 前記判別手段は、前記レベルの個数が2つかどうかを判別することを特徴とする請求項7に記載の画像処理装置。

【請求項10】 前記符号化手段による符号化は、前記画素値のレベルを表す情報と、前記画素値の連続を表す情報を出力することを特徴とする請求項7に記載の画像処理装置。

【請求項11】 画像を多値画像として入力する入力ステップと、

前記多値画像をブロックに分割する分割ステップと、前記分割ステップで分割されたブロック内の画素値のレベルの個数が所定の数かどうかを判別する判別ステップと、

前記判別ステップにおいて、前記レベルの個数が所定の数であると判別された場合は、前記判別ステップで判別されたブロックに対してランレングス符号化を行う符号化ステップとを有することを特徴とする画像処理方法。

【請求項12】 画像を複数の符号化対象ブロックに分割する分割手段と、

前記分割手段による分割される符号化対象ブロックのサイズを選択する選択手段と、

前記複数の符号化対象ブロックの各々を符号化する符号化手段とを有し、

前記選択手段によるサイズの選択は、前記符号化手段により前記画像を完全に符号化することなく行われることを特徴とする画像処理装置。

【請求項13】 画像を複数の符号化対象ブロックに分割する分割ステップと、

前記分割ステップで分割される符号化対象ブロックのサイズを選択する選択ステップと、

前記複数の符号化対象ブロックの各々を符号化する符号化ステップとを有し、

前記選択ステップで選択されるサイズの決定は、前記符号化ステップで前記画像を完全に符号化することなく行われることを特徴とする画像処理方法。

【請求項14】 画像を複数の符号化対象ブロックに分割する分割手段と、

前記分割手段による分割される符号化対象ブロックのサイズを、該符号化対象ブロックの符号量に基づいて前記画像内で可変に設定する設定手段と、

前記複数の符号化対象ブロックの各々を符号化する符号化手段とを有することを特徴とする画像処理装置。

【請求項15】 画像を複数の符号化対象ブロックに分割する分割ステップと、

前記分割ステップで分割される符号化対象ブロックのサイズを、該符号化対象ブロックの符号量に基づいて前記画像内で可変に設定する設定ステップと、

前記複数の符号化対象ブロックの各々を符号化する符号化ステップとを有することを特徴とする画像処理方法。

【請求項16】 画像のブロック領域毎に符号化対象ブロックを決定する決定手段と、

前記決定手段により決定される符号化対象ブロックのサイズを前記ブロック領域毎に該ブロック領域内で可変に設定する設定手段と、

前記複数の符号化対象ブロックの各々を符号化する符号化手段とを有することを特徴とする画像処理装置。

【請求項17】 画像のブロック領域毎に符号化対象ブロックを決定する決定ステップと、

前記決定ステップで決定される符号化対象ブロックのサイズを前記ブロック領域毎に該ブロック領域内で可変に設定する設定ステップと、

前記複数の符号化対象ブロックの各々を符号化する符号化ステップとを有することを特徴とする画像処理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本願発明は画像の符号化を行う画像処理装置及び方法に関するものである。

【0002】

【従来の技術】従来、多値画像の可逆符号化方式としてDPCM(Differential Pulse Code Modulation)が広く知られている。一般的なDPCMによる符号化は、符号化対象画素の周囲にある符号化済の画素、すなわち左画素(これをaとする)、上画素(同様にb)、左斜め画素(同様にc)を用いて符号化対象画素の予測値(平面予測)を例えば、式 $(a+b-c)$ を用いて算出し、これと符号化対象画素値との差分を符号化するものである。

【0003】写真などの自然画像をスキャンした画像データに代表される多値階調画像は、画像中の部位により統計的性質が大きく異なるものである。統計的差異は、画像中のエッジや輝度勾配の方向等の局所構造の違いにより起こるものである。

【0004】従来例に示したDPCM方式による画素値予測は、これら局所的な構造のうち、部分的に二次元平面に近似されうるものに対してのみ有効であり、他の構造に対しては却ってダイナミックレンジを増やしてしまい逆効果となる。

【0005】これを補う形として、例えば、符号化をブロック単位とし、ブロック内で支配的な構造に対して有効な符号化方法を用い、その方式を示すインデックスをサイド情報として符号化する方法が考えられる。

【0006】しかしながら、画像内の局所構造は、一般に、平坦(低周波構造)部分は面積が広く、輝度変化が急峻なエッジや細線構造では数画素程度の小さな構造をとるものである。

【0007】従ってブロック単位で処理を行う場合、平坦部では大きなブロックを用いるのが有利であり、微細構造部では小さなブロックを用いた方が効率が良いという背反を生じる。

【0008】さらにつけ加えるならば、平坦部分でも、その形状が複雑なものあれば、高周波構造でも広い面積にわたって一様に広がるものもある。以上の様に従来、画像の性質に応じて最適な画像処理単位で最適な符号化方法を用いた符号化を行うことは出来ないという問題があった。

【0009】上述の問題を解決するために本願発明は画像の性質に応じて、最適な画像処理単位で最適な符号化方法を用いた符号化を行うことを目的とする。

【0010】また、ブロック単位の符号化において従来よりも更に効率の良い符号化を行うことを別の目的とする。

【0011】

【課題を解決するための手段】上述の課題を解決するために、本願発明の請求項1の画像処理装置は、画像を複数の符号化対象ブロックに分割する分割手段と、前記分割手段により分割された各々の符号化対象ブロックに対して、複数個用意された符号化モードから前記符号化対象ブロックに用いる符号化モードを1つを選択する選択手段と、前記選択手段により選択された符号化モードに応じて前記各々の符号化対象ブロックを符号化する符号化手段とを有し、前記分割手段による分割は、前記画像内に異なるサイズの符号化対象ブロックの存在を可能とすることを特徴とする。

【0012】

【発明の実施の形態】本願発明の実施の形態の一例を図1及び図2を用いて説明する。

【0013】図1は画像の符号化復号化を行うための回路構成であり、1は画像データの符号化及び復号化をソフトウェア処理で行うことができるCPU制御部、2は画像データを取り込む画像入力部、3は画像データを印刷するプリンタ部、4は画像データをモニタなどに表示する表示部、5はCPU制御部1で符号化されたデータを送信、又は外部機器から送信された符号化データを受信することができる通信制御部、6は画像メモリである。

【0014】まず画像入力部1から入力された画像データ(ここでは8bit階調の輝度画像データとする)

は、一旦画像メモリ6に蓄積される。

【0015】次に画像メモリ6内の画像データに対し、後述する図2のフローチャートの様なソフトウェア処理をCPU制御部1を用いて行い符号化する。

【0016】なお、符号化の方法は、正方ブロック単位で符号化が行われ、基本ブロックサイズは128×128画素の正方ブロックである。また最小単位のブロックサイズは4×4画素の正方ブロックであり、後述するブロックサイズを変化させて符号化出来る様な符号化を行う。

【0017】図2の各ステップでの処理の説明の前に、本実施例で画像のブロック単位に用いる複数の符号化方式について説明する。

【0018】ある符号化単位のブロック（本実施例では4×4、8×8、16×16、32×32、64×64、128×128画素）の注目画素\*について、図3(a)に示す様に周囲画素をa、b、cとする。さらに符号化単位ブロックの左上コーナーの画素値をC0（ゼロ）とする。

【0019】このとき、本実施例における符号化方式は、上述の画素を用いて図3(b)の様に示される。

【0020】符号化方法は全部で9種類を設定し、この内#0番～#5番の6種類の符号化方法は周辺画素との差分値を符号化する予測符号化方法であり、差分値（予測誤差）を後述するハフマン符号を用いて符号化するものである。

【0021】ハフマンテーブルは同じブロックサイズの\*

$$A = \sum_k x_k^2, B = \sum_k y_k^2, C = \sum_k (C0 - d_k)^2, D = \sum_k (C0 - d_k) x_k,$$

$$E = \sum_k (C0 - d_k) y_k, F = \sum_k x_k y_k$$

とおくと、 $E = Aa^2 + 2Da + Bb^2 + 2Eb + C + 2Fab$ となる。

【0029】これに対して

【0030】

【外3】

$$\frac{\partial}{\partial a} E = 0, \frac{\partial}{\partial b} E = 0$$

として解くことにより、 $a = (BD - FE) / (F^2 - BA)$ 、 $b = (AE - FD) / (F^2 - BA)$ となる。A、B、Fはブロック内の位置が関係するだけであり、kに対して一定の値であるから、画素値 $d_k$ が関係するC、D、Eの値を測定することによりa、bが簡単に求まる。a、bを±0.5の範囲でクリップし、このクリップされた区間を0～255の値にデジタル化する。即ちデジタイズ後のa、bの値を $a'$ 、 $b'$ とすると、 $a' = 128 + \text{floor}(128a + 0.5)$ 、 $b' = 128 + \text{floor}(128b + 0.5)$ となる。

\* #0番～#5番の符号化方法には共通のものを用いるが、ブロックサイズが異なる場合は、予測誤差の統計も異なるので、各々の符号化単位のブロックサイズに適したハフマンテーブルを用いる。よって本実施例では4×4、…、128×128画素のブロックに対応する計6個のハフマンテーブルを有する。

【0022】次に#6番の“平面近似モード”の符号化方法を説明する。

【0023】図3(a)のブロック内のC0の画素位置を3次元(x軸、y軸、z軸)の(O、O、C0)であると仮定し、図の右方向(主走査方向とする)をx軸、図の下方向(副走査とする)をy軸とする。

【0024】ここで符号化単位のブロック全体の画素値に対するZ方向の自乗誤差を求めることにより平面( $z = C0 + ax + by$ )として近似し、a、bを求める。

【0025】ブロック内のK番目画素値を $d_k$ 、そのx、y座標、を $x_k$ 、 $y_k$ とすると、Z方向の自乗誤差のブロック全体の和Eは、

【0026】

【外1】

$$E = \sum_k \{ d_k - (ax_k + by_k + C0) \}^2$$

となる。

【0027】これを展開し、

【0028】

【外2】

【0031】上述で求められたデジタル値 $a'$ 、 $b'$ を用いて予測誤差 $e_k$ を求めると、

$$e_k = d_k - \{ \{ x_k (a' - 128) + y_k (b' - 128) \} / 256 + C0 \}$$

【0032】以上の様にして最適近似平面を定義するパラメータa及びbが求まり、この平面との予測誤差が求まる。

【0033】次に求められた値をいかに符号化するかにについて説明する。

【0034】通常符号データは図4(a)～(d)に示す様な構造となる。

【0035】即ち符号化対象のブロックに対して#6番の“平面近似モード”が用いられたことを示すハフマン符号と $a'$ 、 $b'$ の値(固定長)と、C0のもしくはC0' (C0'は前の符号化対象のブロックC0との差分)との符号を送る。

【0036】次に完全近似フラグデータを送る。完全近似フラグとは、符号化対象のブロック内の画素に対する予測誤差 $e_k$ が全て0になったかどうかを示すフラグであり、全て0ならばフラグオン（本実施例では1）にしてデータを終える。そうでなければフラグオフ（本実施例では0）にして、その後、上述で発生した予測誤差 $e_k$ をハフマン符号化したデータを $e_k$ から順に送る。

【0037】ここで $e_k$ の符号化用いるハフマンテーブルと#0番～#5番の符号化方法に用いるハフマンテーブルは同じものを用いることにし、このテーブルは図2のS1及びS7について後述されるハフマンテーブルである。

【0038】ここでC0とC0'のいずれを用いるかは、傾きのパラメータ $a'$ 、 $b'$ によって異なる。 $a'$ 、 $b'$ に対しては各々にしきい値を設定する。本例ではこの値を16とし、0～255の値をとる $a'$ 、 $b'$ の値が両方とも $128 \pm 16$ （本実施例の128は通常のグラフの傾き0に対応）の範囲に収まった時、即ち $a'$ 、 $b'$ とも傾きが小さい時にC0'が小さくなる可能性があるのでC0'を用いることとする。それ以外の場合にはC0を用いる。

【0039】C0は固定長で送るがC0'の場合はハフマン符号を用いる。このC0'に用いるハフマンテーブルは固定であり、本実施例では図3（e）のテーブルを用いる。

【0040】次に図3（b）における#7番の“一様”の符号化方法について説明する。この符号化方法はブロック内の全データが同一画素値である時に用いられ、画像に枠がついていたり、CG（コンピュータグラフィック）の背景等に対して有効である。

【0041】符号データは図4（b）に示す様になり、符号化方法を示すハフマン符号と、#6番の符号化方法で用いたC0'と同じ性質をもつC0'の符号とを送る。

【0042】次に#8番の“二値モード”の符号化方法について説明する。

【0043】これは、符号化対象のブロック内に2種類の画素値しか存在しない場合に用いる方法である。

【0044】2種類の画素値の内、輝度の高い方の画素値を $D_1$ 、他方を $D_2$ とした時に、最初の画素を $D_1$ を仮定して、図5（a）に示す様な画像スキャンを行うことにより $D_1$ 、 $D_2$ が交互に発生する場合のランレングスを符号化する。

【0045】この得られたランレングスに対して予め用意されたハフマン符号化を行う。1つのランレングスデータは、0個以上のMake up Codeと1つのTerminating Codeとを用いて表される。

【0046】Terminating Codeはランレングスが0～127までを表すコードである。Make

up Codeは128以上の大きな値を表すコードである。これらのコードは各々に用意されたハフマンテーブルにより符号化される。

【0047】ブロック中の最後のランはそのまま符号化されずEOBのコードを付加して処理される。 $D_1$ 、 $D_2$ を例えば白、黒とすると、ハフマンテーブルは白、黒で別個に持つものとし、固定のテーブルとする。

【0048】以上で本実施例に用いられる9種類の符号化方法についての説明を終える。なお、この複数の符号化方法は、符号化対象のブロックサイズを変化させて符号化できる後述の符号化方法において、各々の異なる領域のブロックに対して、上記9種類の内の最適な符号化方法を選択して割り当てながら、1画面内の符号化方法の決定処理が行われる。

【0049】詳細は後述する。

【0050】次に符号化処理全体の流れを表す図2の各ステップの処理について詳細に説明する。

【0051】ステップS1ではステップS3の処理の際前述の#0～#6の符号化方法に用いる基準のハフマンテーブルを各ブロックサイズに対して1つ設定する。

【0052】ステップS2では、画像メモリ6に蓄積された画像データを、基本ブロックである $128 \times 128$ 画素単位に分割し、この基本ブロック毎に順次取り出す。

【0053】ステップS3では、画像データをステップS9で最終的に符号化する際のブロック（符号化対象ブロックとする）への分割方法、及び分割された各々のブロックに対する最適符号化方法（#0～#8）を仮決定する。以下、詳細に説明する。

【0054】符号化対象ブロック（ブロック分割方法）は1つの正方ブロックを符号化する際に発生する符号量と、該正方ブロックを4分割した4つのサブブロックを符号化する際に発生する総符号量とを比較した結果に基づいて仮決定される。

【0055】あるサイズの正方ブロックとそのサブブロックの様子を図6に示す。

【0056】 $2^n \times 2^n$ 画素の正方ブロックをB、これを4分割して得られる $2^{n-1} \times 2^{n-1}$ 画素のサブブロックを $b_0 \sim b_3$ とする。

【0057】1つの正方ブロックBに対し、前述の#0～#8の9種類の符号化方法のうち、#iを用いた時の符号化データの符号量を求める関数を $C_i$ とし、符号量を $C_i(\beta)$ と書くことにする。

【0058】また、9種類の符号化方法を用いて符号化された符号化データの符号量のうち、最小の符号量を求める関数をBWとすると、 $BW(\beta) = \min(C_i(\beta))$ と書ける。

【0059】また、あるサイズのブロックに対する最小符号量を求める関数をMCとすると、この時、前述のB、 $b_i$ 等を用いて以下の様に書ける。

10

20

30

40

50



【0060】

\* \* 【外4】

$$MC(B) = \min(\{BW(B) + HD\}, \sum_{i=0}^3 MC(b_i) + SF) \\ / BW(B) = \min(C_i(B))$$

【0061】本実施例では最小サブブロックのサイズを  
4×4画素としているので、上式及び

※ 【0062】  
※ 【外5】

$$MC(B_{4 \times 4}) = BW(B_{4 \times 4}) = \min(C_i(B_{4 \times 4}))$$

となる。ここでSFは、サブブロックへの分割の有無を表すためのフラグのデータ量であり、本実施例では、1つのブロック、サブブロックを更に4分割するかどうかを表せれば良く、1bitの符号を要する。

【0063】従ってSF=1となる。

【0064】また、HDはBW関数によって選ばれたiを表す為の情報量であり本実施例ではiの情報を専用のハフマンテーブルでハフマン符号化する。

【0065】符号化対象ブロックのサイズを仮決定する方法を簡単に説明すると、上述の式を用いてある正方ブロックの符号量とこのサブブロックの総符号量のうち符号量の少ない方のブロックサイズを選択し、この結果サブブロックの方を選択した場合には、各々のサブブロックを上述の正方ブロックに置き換えて、更にブロックサイズの選択を行うものである。また、各ブロックの符号量は、9種類の符号化方法の1番相性の良い方法を用いて算出されたものである。

【0066】上記の式によって示されるブロック分割方法及び分割された各ブロックの符号化方法の仮安定の様子を図7に示す。

【0067】図7のブロック分割例は、128×128画素を基本ブロック（符号化対象ブロックの最大サイズ）とし、再帰的にブロックを4分割して符号化対象ブロックを仮決定した結果である。

【0068】実際には、128×128画素ブロック内を右上から左下にかけてエッジが横断している場合に図のようなブロック分割が起こる。また、分割された符号化対象ブロックの各々は独立の符号化方法（#0～#8番）を有している。

【0069】図8は、図6のブロック分割例に基づいて記述されたブロック分割方法及び分割されたブロック（符号化対象ブロック）各々の符号化方法を示すデータの形態を表している。

【0070】図中P<sub>n</sub>は前述したSF（ブロック分割の有無）に対応する1ビットのデータ（ブロック分割データとする）であり、P<sub>n</sub>=1ならばそのブロックを分割、P<sub>n</sub>=0ならばブロック分割せずに符号化方法を示すデータを続ける。（nは処理対象のブロックサイズを示す。）

【0071】例えば、（P<sub>1,2</sub>=1）、（P<sub>4,4</sub>=1）、（P<sub>1,2</sub>=0）とビットが続いた場合には（128×128画素ブロックが4分割）、（左上の64×64画素ブ

ロックを4分割）、（左上の32×32画素ブロック分割せず）と判断し、符号化方法を表すハフマンデータhd（前述のHDに対応）を付ける。更に続いて（P<sub>1,2</sub>=1）、…と続く。

【0072】P<sub>n</sub>=1ならば続けてブロック分割データP<sub>n+1</sub>が続く。

【0073】また図8において、最小ブロックサイズ符号化対象ブロックB<sub>1</sub>～B<sub>8</sub>、B<sub>9</sub>～B<sub>12</sub>に対応するデータには、ブロック分割データは付けない。これは最小ブロックを分割することがないので必ずP<sub>n</sub>=0となるからである。

【0074】符号化対象ブロック※から※※までのブロック分割データ及び符号化方法を示すデータは省略する。

【0075】ステップS<sub>1</sub>では、図8に示したブロック分割データ及び符号化方法を示すデータを保存しておく。

【0076】ステップS<sub>1</sub>では、ステップS<sub>1</sub>で分割された図7に示す各々の符号化対象ブロックの中で、符号化方法の#0番～#6番までを用いて符号化されることが仮決定されているブロックに対して、符号化時に発生する差分値（予測誤差）の統計を各ブロックサイズ毎にとる処理を行う。

【0077】例えば図7において、B<sub>1</sub>とB<sub>1,1</sub>、又はB<sub>1</sub>とB<sub>1,2</sub>、又はB<sub>9</sub>～B<sub>8</sub>とB<sub>4</sub>～B<sub>1,2</sub>は、同じブロックサイズなので、まとめて統計がとられ、1画面分の統計を蓄積する。

【0078】本実施例では予測誤差-255～255の計511個のカウンターが各々のブロックサイズに必要なので計6×511個のカウンターを用いる。

【0079】ステップS<sub>6</sub>ではステップS<sub>2</sub>～ステップS<sub>5</sub>までの128×128画素単位のブロック処理を1画面分行ったらステップS<sub>7</sub>へ、1画面の途中ならばステップS<sub>2</sub>にもどり次のブロックの処理に進む。

【0080】ステップS<sub>7</sub>では、ステップS<sub>5</sub>でとられた6つの統計に基づいて、各ブロックサイズに最適なハフマンテーブル（最適ハフマンテーブル）が作成される。

【0081】ステップS<sub>8</sub>では予め設定しておいた最適化処理のループ数を満たしたか否かが判定され、ループ数を満たしていればステップS<sub>9</sub>へ、満たしていなければステップS<sub>1</sub>へもどり、前述で作成されたハフマンテ

ーブルをステップS1の基準ハフマンテーブルとして書き換える。これは基準ハフマンテーブルよりも符号化対象の画像に基づいて作成された最適ハフマンテーブルを用いて符号化対象ブロックを生成する為のブロック分割、及び符号化方法を決定し直すことにより、更に効率の良い符号化が行えるからである。

【0082】ステップS8で設定回数のループを行ったら、最後のループで決定されたブロック分割方法及び符号化方法、ならびに作成された最適ハフマンテーブルを最終的な符号化に用いる参照情報として保存し、ステップS9へ進む。

【0083】ステップS9では参照情報を元に画像の符号化を行い符号化データを出力する。

【0084】図9はステップS9での符号化処理の流れを示したものである。同図に於いてS80は最適ハフマンテーブルを符号化するステップ、S81は基本ブロックのサイズのブロックデータを取り出すステップ、S82はこのブロックデータに対応する参照情報を取り出すステップ、S83は参照情報に従ってこのブロックを符号化するステップ、S84は画像の終端を判定するステップである。

【0085】まず、ステップS80では128×128画素ブロック用ハフマンテーブル64×64画素ブロック用ハフマンテーブル、32×32画素ブロック用ハフマンテーブル、16×16画素ブロック用ハフマンテーブル、8×8画素ブロック用ハフマンテーブル、4×4画素ブロック用ハフマンテーブルを符号化して出力する。

【0086】ステップS81では、画像データから順に基本ブロックのサイズでブロックデータを取り出す。

【0087】ステップS82では、このブロックデータに対する参照情報を呼び出す。ステップS83で、参照情報とこの参照情報に従ってブロックデータを符号化して生成した符号化データ列を出力する。ステップS84では画像の終端を判定し、未処理のブロックがある場合には次のブロックについてステップS81から処理を行う。

【0088】以上の処理により、図10に示す様な画像データに対する符号化データ列を生成し、出力する。なお、データの順番は、1画素単位でなくブロック単位で送る様にしても良い。以上で第1の実施の形態の説明を終る。

【0089】また、ブロック分割の方法としてブロックを同サイズに4分割する方法を挙げたが、ブロックサイズを可変にしても良い。

【0090】また、第1の実施の形態では図6の様にある正方ブロックの符号量とこれを4分割した複数のサブブロック符号量とに基づいて、ブロック分割して符号化対象ブロックを決定していたが、最大ブロック(128×128画素)から最小ブロック(4×4)までの全て

のブロックについての符号量を求めてこれら全ての符号量に基づいてブロック分割しても良い。

【0091】また、初めに最小ブロックの符号量を求め、このブロックをサブブロックと見なし、図6の様に4つのサブブロック(4×4画素)の総符号量とこれらを合わせた領域のブロック(8×8画素)の符号量とに基づき仮のブロック分割方法と符号量を決定し、4つの決定された領域(8×8画素)の総符号量と決定された領域(8×8画素)から構成される更に大きなブロック(16×16画素)の符号量とに基づき新たに仮のブロック分割方法を決定し、最大ブロックサイズ128×128画素までこれを繰り返すことにより符号化対象ブロックを決定する様にしても良い。

【0092】また、選択可能なブロック符号化方法として予測符号化方法と平面符号化方法、ランレングス符号化方法を用いたが、予測方法を変えるなどして適用方法を変えたり、あるいはマルコフモデル符号化方法等、別の符号化方法を用いても良いことは言うまでもない。更に、上述の実施の形態では各符号化方法におけるエントロピー符号方法としてハフマン符号化を用いているが、勿論これは算術符号化等、他のエントロピー符号化方法を用いてもよい。

【0093】またブロック分割方法の決定は、符号量に基づいて行われていたが、実際に符号化することにより符号化効率を求め、この符号化効率に基づいて行う様にしても良い。

【0094】

【発明の効果】以上のように本願発明によれば、画像の性質に応じて、最適な画像処理単位で最適な符号化方法を用いた符号化を行うことができる。

【0095】また、ブロック単位の符号化において従来よりも更に効率の良い符号化を行うことができる。

【図面の簡単な説明】

【図1】本願発明の実施例に用いる装置の図。

【図2】本願発明の実施例の全体の流れを表すフローチャート。

【図3】実施例で用いる符号化方法を説明する図。

【図4】#6、#7の符号化方法を用いた際の符号化データ形態及び該符号化方法に用いるC0'に対するハフマンテーブルを示す図。

【図5】二値符号化方法で用いるブロック内のスキャン方法と符号化データ形態を説明する図及び二値符号化で用いるランレングスの一覧表を示す図。

【図6】ある正方ブロックとそのサブブロックを表す図。

【図7】符号化対象ブロックを求める為に基本ブロックを更にブロック分割した様子を示す図。

【図8】ブロック分割情報の形態を表す図。

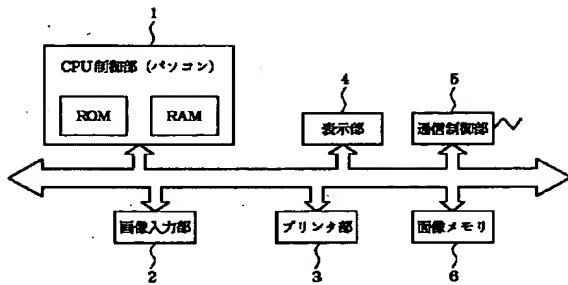
【図9】S9の符号化処理についてのフローチャート。

【図10】最終的に符号化された画像データを示す図。

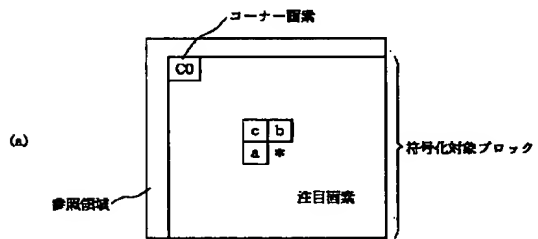
【符号の説明】

\* \* 1 CPU制御部

【図1】



【図3】

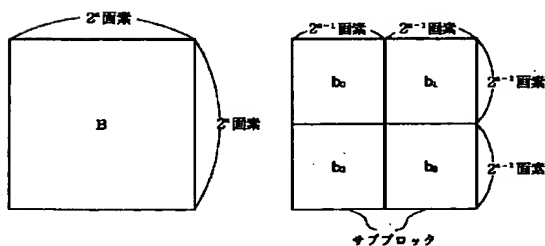


(b)

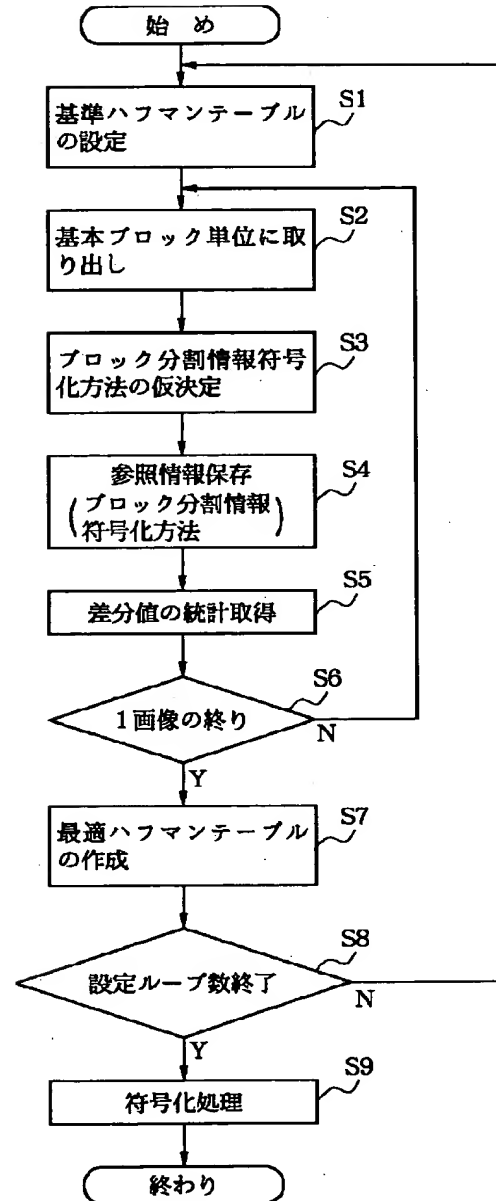
番号	符号化方法
#0	$a + b - c$
#1	$b$
#2	$a$
#3	$b + (a - c) / 2$
#4	$a + (b - c) / 2$
#5	$(a + b + c) / 3$
#6	平面近似モード
#7	一様モード
#8	二値モード

ブロックサイズ毎にハフマンテーブル共有

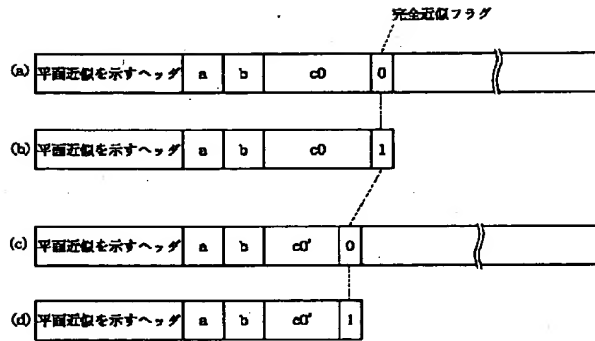
【図6】



【図2】



【図4】

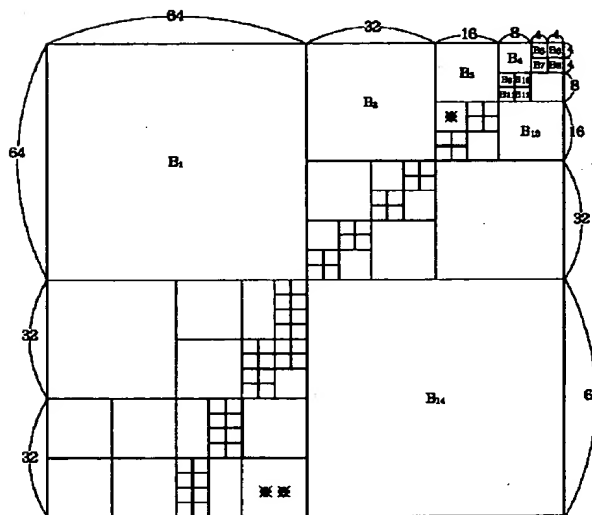


(a)

値	コード	コード長
0	0	1
1	1 0 0 S	4
2	1 0 1 S	
3	1 1 0 S	
4	1 1 1 0 0 0 S	7
5	1 1 1 0 0 1 S	
6	1 1 1 0 1 0 S	
7	1 1 1 0 1 1 S	
8	1 1 1 1 0 0 S	
9	1 1 1 1 0 1 S	
10	1 1 1 1 1 0 S	
11	1 1 1 1 1 1 0 0 S	9
12	1 1 1 1 1 1 0 1 S	
13	1 1 1 1 1 1 1 0 S	
ESC	1 1 1 1 1 1 1 1	8+8

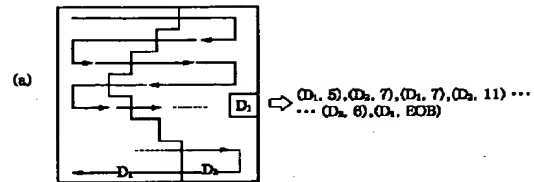
c0' ハフマンテーブル

【図7】



符号化対象ブロックの分割例

【図5】



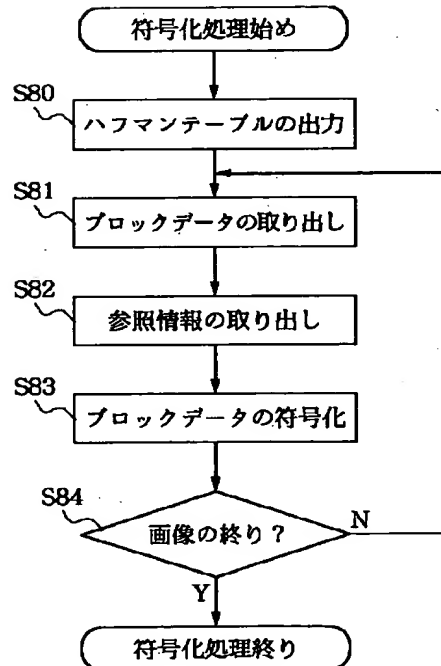
二値モード ランレングス表

(b)

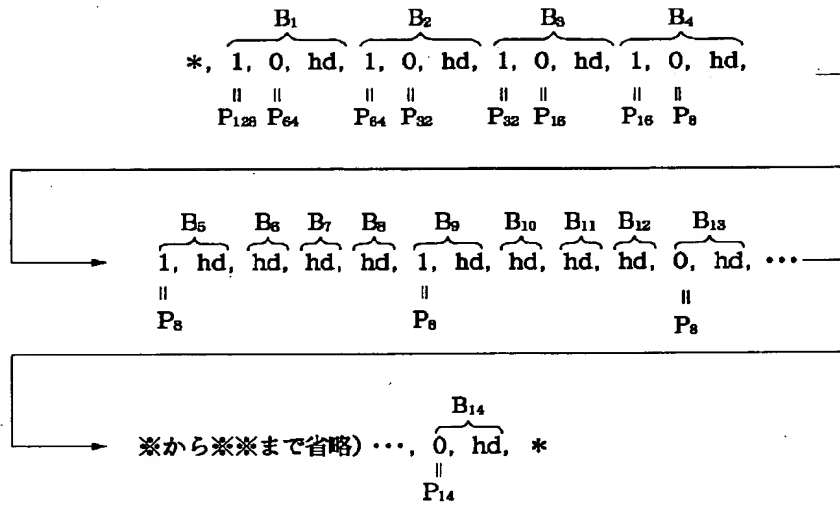
Terminating Code	Make-Up Code
0	128
	256
	512
	1024
	2048
127	4096
EOB	8192

\* EOB: End Of Block

【図9】



【図8】



【図10】

最適 ハフマン テーブル データ	参照情報 + $\alpha$ (ブロック分割情報 + 符号化方法 + その他のパラメータデータ)	差分値 (予測誤差) データ
---------------------------	---	----------------------

1 画像